



DIME Dynamic Documentation Training Stata Exercise

Luiza Andrade & Mrijan Rimal

June 8, 2017

For the most recent version of this file, please check <https://github.com/worldbank/DIME-LaTeX-Templates/>

Introduction

This exercise introduces you to how to export files from Stata that can be read in L^AT_EX. See exercises 1 and 2 for instructions on how to import files into L^AT_EX. After this exercise and exercise 1, you will have a document that is automatically updated each time you run your Stata and your L^AT_EX code.

We have provided you with a do-file that creates files you can import to a L^AT_EX document. We will go through these examples and then we will ask you to create some tables and graphs of your own using your own data. Note that this is not an exercise on Stata, only on exporting Stata outputs into format that L^AT_EX can read, so this exercise assumes knowledge of some intermediate level Stata commands.

Part 1: Setting up a Folder Structure

Do not rush over this part. You will be forced to write an unnecessarily complicated L^AT_EX code unless you set up a simple but well organized folder structure where you'll save tables and figures created in Stata. We strongly recommend that you start by setting up the following folder structure.

Create one folder called **Output**, and inside this folder, create one folder called **Raw** and one folder called **Final**. You will export tables and graphs from Stata to the **Raw** folder and then import them to your L^AT_EX document. When you start a new L^AT_EX, you must save the **.tex** file in the **Final** folder before compiling it, so L^AT_EX knows where the file paths specified lead.

See the example below. As your project grows bigger, it is common that sub-folders are added in the **Raw** folder. For example, tables and figures often have separate folders. You will find your preferred way to organize this.

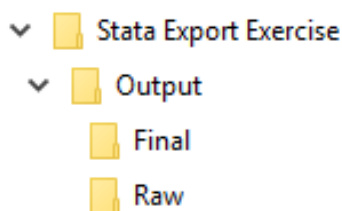


Figure 1: Common Folder Structure

Part 2: Setting your path for Stata

We now need to tell all commands in Stata to save the tables and figures they export to the **Raw** folder. We want to create a global with the file path pointing to the **Raw** folder that we can use in our commands. Using a global instead of typing out the folder location in all commands both makes the code simpler, and makes it easier to update if you would have to move your folders.

You will find a do-file called **Export tables and images.do** in the same folder as this handout. Open it

and look for the part that says:

```
global main_folder "<<<ENTER YOUR FOLDER PATH HERE>>>"
```

You need to enter the path to the folder you created in part 1 here. If you have a Windows computer and you created your **Output** folder in your Documents folder, then your file path should look similar to this:

```
global main_folder "C:\Users\JoeSmith\Documents\Output"
```

The following two sub-sections help you find the file path to the folder you just created if you are not sure how to do that. The first sub-section gives advice for Windows, and the second for Mac.

Finding Path on a Windows Computer

Path to a file can be found by selecting a file and pressing **SHIFT on your keyboard and RIGHT CLICKING your mouse and then clicking COPY AS PATH in the resulting drop down menu** as shown in Figure 2.

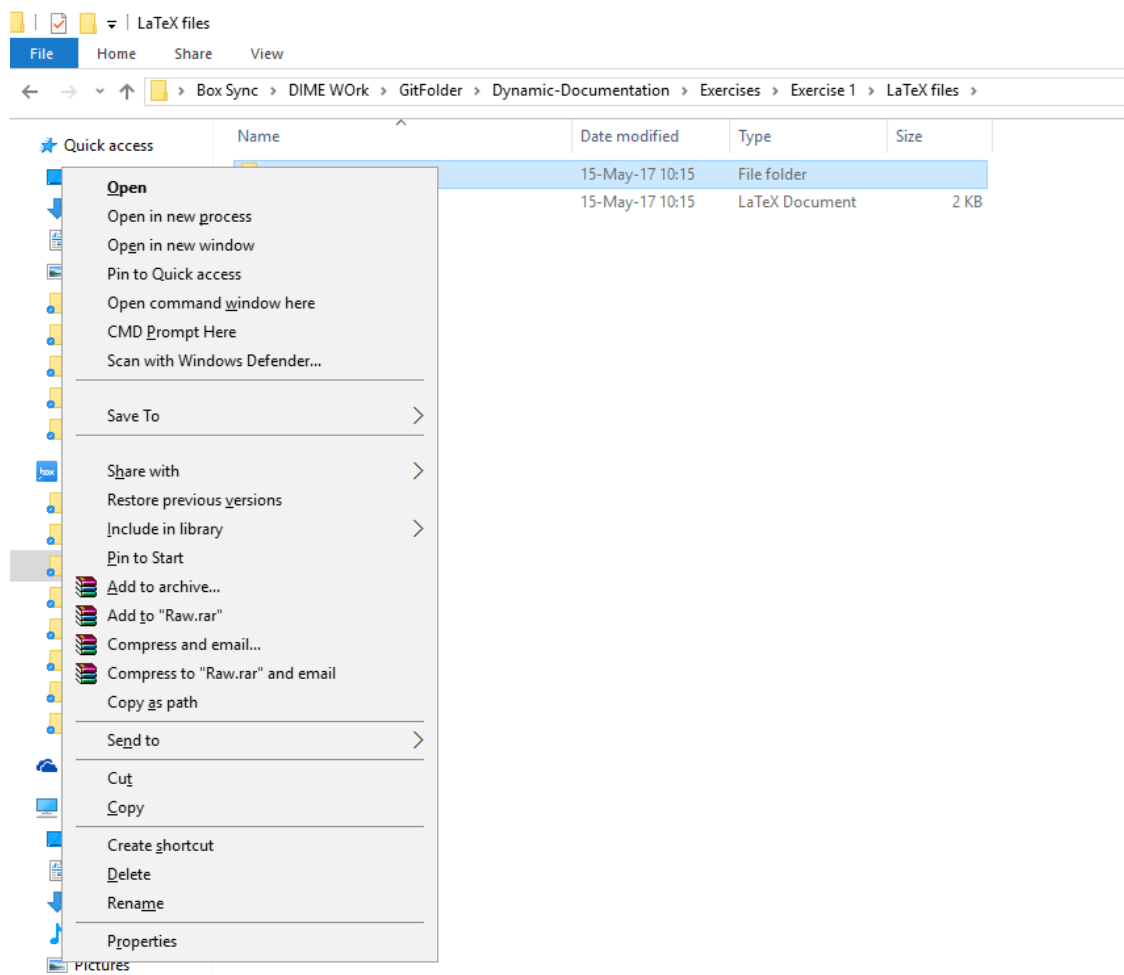


Figure 2: Finding path on a Windows Computer - Solution 1

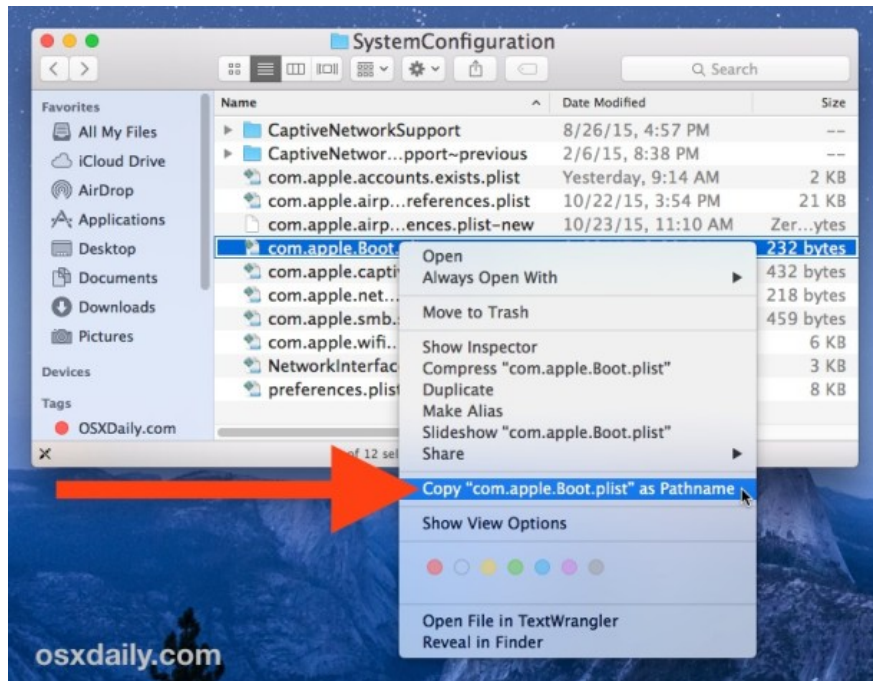


Figure 3: Finding a file path on a mac. Photo courtesy osxdaily.com

Another solution to finding the path on a Windows computer is shown in Annex 1.

Finding Path on a Mac

To copy the path on a Mac, please follow the following steps:

- Right click(or Control-click or two finger click) on the file or folder you want to copy the path of.
- While the right click menu appears, press the **Options** key to reveal ‘‘Copy file as pathname’’ button. This is shown in figure 3.
- Click on that button and paste anywhere to paste the path name.

If you’re using an older MAC computer and the option to “Copy file as pathname” does not show up, please follow the following steps:

- Right click(or control-click or two finger click) on your file or folder you want to copy the path of.
- Click on **Get Info** option that appears in the drop down menu.
- In the **Info** window, copy the part that appears after **Where:** and before **Created:**.
- Paste that part anywhere to paste the path name.

Part 3: File Formats and Names

Not all file formats that you can chose when exporting from Stata can be imported into L^AT_EX. Figures must be saved in `.png` format. All figures produced in Stata can be exported in `.png`. Even if some commands are not able to export in this format, you can use Stata to convert the figure for you. You will find more details on this below, when we export figures.

Tables must be saved in `.tex` format. Tables cannot be converted as easily as figures, so it is usually easier to find an alternative command that produces the same table but can export to `.tex` format. Most commands that export tables are able to export to this format so it should not be difficult to find an alternative.

Files saved in MS Word, MS Excel, or Stata's `.gph` format cannot imported to L^AT_EX.

Files exported from Stata should have a descriptive name. Otherwise, you increase the risk for human errors, and the whole point of dynamic documents is to reduce exactly that. For example – rather than `graph1.png`, `graph2.tex`, names like `treatmentEffectGraph.png` would make it easier to understand what files we are using.

Part 4, Task 1: Tabulate Categorical Variables

The codes for Part 4, 5 and 6, are all in the do file `Export tables and images.do`. The data you will use is a sample data set that all instances of Stata already have, so you do not need a data file, as the do-file will load this data set for you.

First we want to start by exporting a simple frequencies table. We first use the command `tabulate` to generate the statistics, and then use `esttab` to export it to `.tex` format. The command `estpost` that comes before the `tabulate` command adjusts the result of the tabulation to a format that `esttab` can use. See the help files for `estpost` and `esttab` for more details on how this works. This is an exercise in L^AT_EX and not in the `estout` package, so we will not go into more details about it.

To export the tabulation to `.tex` format using `esttab`, you simply set the file extension in the `using` option. Like this: `using "$raw_output/categorical.tex", replace`. Remember to always have an explanatory file name.

Now check your `Raw` folder to see the `categorical.tex` file you have exported.

Part 4, Task 2: Regression table

Similar to the previous task, we can use `esttab` to export regression tables. First we run the regressions whose results we want to export. The command `eststo:`, when typed before the regression commands, stores regressions' result so that `esttab` can export a table with results for more then one regression at a time. You can use any regression options allowed by Stata. If Stata can run the regression and display its results in the Stata window, then `esttab` will be able to export them to `.tex` format.

The difference between the regressions in lines 109 and 113 is that the second one includes region fixed effects. We can use the command `estadd` to add text to the tables. In this example, we use it to indicate the inclusion of fixed effects. See the help files for `estout`, `estadd` and `esttab` for more details on how this works.

Similarly to the previous task, we export the stored regression results using `esttab` to `.tex` format by setting the file extension in option `using`. Like this: `using "$raw_output/regression_table.tex"`. `esttab` will export all results stored in memory as estimates. That is why it is important that we start each task with the code `estimates clear`: otherwise we might add results from previous tables to this table.

Now check your `Raw` folder to see the `regression_table.tex` file you have exported.

Part 5, Task 1: Manually Create a Graph and then Export it

This task shows us how to export graphs created in Stata to a format that L^AT_EX can read. Stata's save graph feature saves the graph in `.gph` format, which only Stata can read. However, the `graph export` feature saves the graph in a picture format, i.e. `png` format, which can be read by the photo viewer on your computer, phone and also by L^AT_EX. It is *absolutely critical* to export the graph to this format, so that L^AT_EX can import it.

That's what line `graph export "$raw_output/regular_graph.png", width(5000) replace` is doing.

Note: This is different from the `iegraph` command, where `save` can both export the graph in `.gph` and `.png` format. For Stata's `graph` command, `export` has to be used to make it readable in L^AT_EX.

Now check your `Raw` folder to see the `regular_graph.png` file you have exported.

Part 5, Task 2: Using `iegraph` to create a figure

Stata's `graph twoway` uses the `save` feature to export pictures in a `.gph` format, and we have to use `graph export` to export it to `.png` format. However, many commands, `iegraph` for example, can export directly to either format using the same `save` option.

This exercise teaches how to use `iegraph` to create a figure and export it to the graphs folder. When you use `iegraph` to export a plot to L^AT_EX, always make sure that the picture has the extension `.png` at the end of the filename. **Without that, `iegraph` is just going to export the picture in a format which only Stata can read!**

Using `save("$raw_output/iegraph.png")` ensures that the graphs are directly saved to the specified output folder.

Now check your `Raw` folder to see the `iegraph.png` file you have exported.

Part 6: Making a Dynamic Document

Here, we will produce a dynamic document. Please only do this if you have completed up to **Part 5, Task 2** of this exercise.

1. Compile the document you've created so far. This will save a PDF file in your **Final** folder with the same name as your `.tex` file.
2. Change the name of the PDF you just created, so next time you compile your `.tex` file, it doesn't save over it.
3. Go to **line 60** of the Stata do file and change the seed from 215320 to a number different than 215320.
Note: Changing seeds is not recommended for actual projects. The change in seeds here is used to highlight the change in treatment/control groups.
4. Rerun the do-file with the new seed.
5. Open your `.tex` file and press *Build and Compile* under *Tools*.
6. If you compare the PDF file you have just generated with the one you saved earlier, you will find that the tables have been updated.

Part 7, Using a do-file to edit a .tex file after exporting it

During this part of the exercise, you will learn how to use commands in Stata to format your tables. While tables exported from Stata to L^AT_EX are generally very nice-looking, sometimes they need to be tweaked a little to look exactly the way you want. In this exercise, you'll use the `filefilter` command in Stata to make small changes to the files exported by Stata.

Annex 1

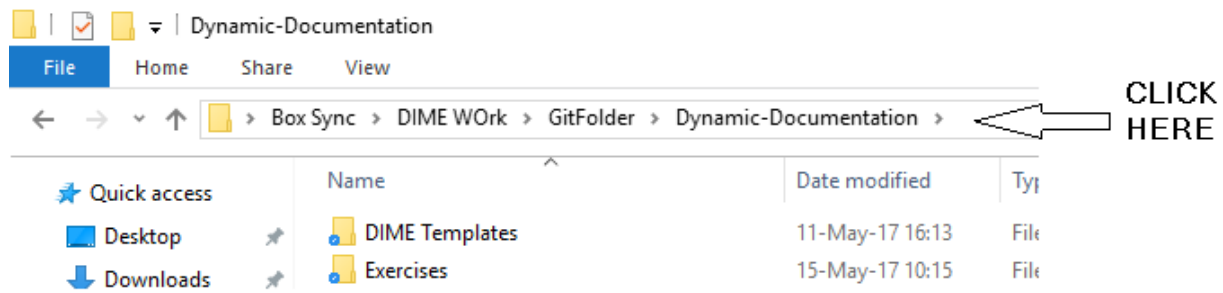


Figure 4: Finding Path on a Windows Computer

As shown in Figure 4, left clicking(normal click) on the bar at the top of the **File Explorer** windows where our files are saved shows us the complete path to the files in a Windows computer.

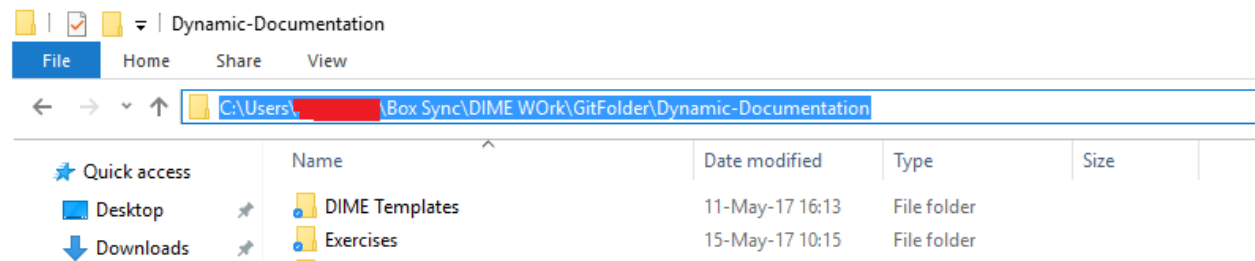


Figure 5: Path shown on a Windows Computer

We can see in Figure 5, that the complete path to the folder is shown. We can then paste this path when setting the path in our Stata do-file and changing the path where it says

```
global main_folder '<<<ENTER YOUR FOLDER PATH HERE>>>'
```